



## CONTROLLER SERVICES

- › DebuggerService
- › ModulesService
- › InterceptorService
- › ExceptionService
- › PluginService
- › HandlerService
- › RequestService

## SAMPLE HANDLER CODE

```
component{  
  
    any function index(event){  
        return "Hello";  
    }  
  
}
```

## CORE PLUGINS

- › AntiSamy
- › ApplicationStorage
- › BeanFactory
- › CFCViewer
- › ClientStorage
- › ClusterStorage
- › CookieStorage
- › DateUtils
- › FeedGenerator
- › FeedReader
- › FileUtils
- › i18n
- › IOC
- › JavaLoader
- › JSON
- › JVMUtils
- › Logger
- › MailService
- › MessageBox
- › ORMService
- › QueryHelper
- › Renderer
- › ResourceBundle
- › SessionStorage
- › StringBuffer
- › Timer
- › Utilities
- › Validator
- › Webservices
- › XMLConverter
- › Zip

## CORE INTERCEPTORS

- › Autowire
- › ColdboxSideBar
- › Deploy
- › EnvironmentControl
- › Security
- › SES

## HANDLER PROPERTIES

- › This scope properties
- › event\_cache\_suffix
- › preHandler\_only
- › preHandler\_except
- › postHandler\_only
- › postHandler\_except
- › allowedMethods

## IMPLICIT HANDLER EVENTS

- › preHandler,pre{Action}
- › postHandler,post{Action}
- › aroundHandler
- › around{Action}
- › onMissingAction

## FOLDER CONVENTIONS

- › Application.cfc
- › index.cfm (empty)
- › config
  - › Coldbox.cfc
  - › Routes.cfm
- › handlers
- › model
- › modules
- › layouts
- › views

## FRAMEWORKSUPERTYPE

- › \$abort(), \$dump(), \$htmlhead(),\$include(),\$throw(),\$rethrow()
- › addAsset(assetList)
- › announceInterception(state,[interceptData])
- › getColdBoxOCM([cachename])
- › getController()
- › getDatasource(alias)
- › getDebugMode()
- › getfwLocale()
- › getInterceptor(name)
- › getModel(name,[ds],[initArguments])
- › getMailSettings()
- › getMailService()
- › getNewMail()
- › getMyPlugin(plugin,[newInstance])
- › getPlugin(plugin,[customPlugin],[newInstance])
- › getResource(resource,[default],[locale])
- › getSetting(name,[fwSetting])
- › getSettingsBean()
- › getSettingStructure([fwSetting],[DeepCopyFlag])
- › includeUDF(udfLibrary)
- › persistVariables(persist,[persistStruct])
- › populateModel(model,scope,trustedSetter,include,exclude)
- › renderExternalView(view)
- › renderLayout(layout)
- › renderView([view],[cache],[Timeout],[LastAccessTimeout])
- › runEvent(event,[prepostExempt],[private])
- › setDebugMode(mode)
- › setNextEvent(event,[queryString],[addToken],[persist],[persistStruct],[ssl],[baseURL],[postProcessExempt],[URL],[URI],[statusCode])
- › setSetting(name,value)
- › setFWLocale(locale)
- › settingExists(name,[fwSetting])

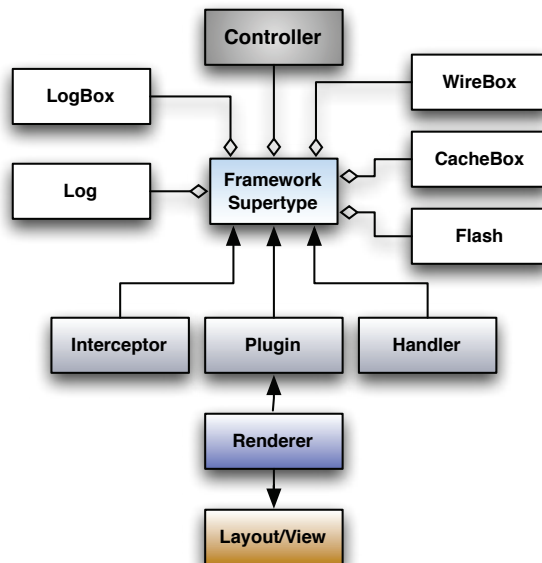
## INTERCEPTOR METHODS

- › configure()
- › getProperties(properties)
- › setProperties()
- › getProperty(name)
- › setProperty(name,value)
- › propertyExists(name)
- › unregister(state)
- › clearBuffer()
- › appendToBuffer(string)
- › getBufferString()
- › getBufferObject()

## CONTROLLER COMMON METHODS

- › getAppRootPath()
- › getCacheBox()
- › getLogBox()
- › getWireBox()
- › getColdboxOCM([name])
- › getColdboxSettings()
- › getConfigSettings()
- › getPlugin(plugin,[customPlugin],[newInstance])
- › get{ServiceName}()
- › getSettingStructure([fwSetting],[DeepCopyFlag])
- › getSetting(name,[fwSetting])
- › persistVariables(persist,[persistStruct])
- › runEvent(event,[prepostExempt],[private])
- › settingExists(name,[fwSetting])
- › setSetting(name,value)
- › setNextEvent(event,[queryString],[addToken],[persist],[persistStruct],[ssl],[baseURL],[postProcessExempt],[URL],[URI],[statusCode])

## COLDBOX COMMON CLASSES



## REQUEST CONTEXT COMMON METHODS (RECEIVED AS EVENT TO EVENT HANDLERS)

- › buildLink(linkto,[translate],[ssl],[baseURL],[queryString])
- › clearCollection()
- › collectionAppend(collection,[overwrite])
- › getCollection([DeepCopyFlag],[private])
- › getCurrentAction()
- › getCurrentEvent()
- › getCurrentHandler()
- › getCurrentLayout()
- › getCurrentModule()
- › getCurrentRoute()
- › getCurrentRoutedURL()
- › getCurrentView()
- › getDefaultLayout()
- › getDefaultView()
- › getEventName()
- › getHTTPHeader(header,[default])
- › getHTTPMethod()
- › getModuleRoot()
- › getSESBaseURL(), setSESBaseURL()
- › getSize()
- › getTrimValue(name,[default],[private]), getValue(name,[default],[private])
- › isAjax()
- › isProxyRequest()
- › isSES()
- › isSSL()
- › noExecution()
- › noRender([remove])
- › overrideEvent(event)
- › paramValue(name,value,[private])
- › renderData( type,data,[contentType],[encoding],[statusCode],[statusText],[jsonCase],[jsonQueryFormat],[jsonAsText],[xmlColumnList],[xmlUseCDATA],[xmlListDelimiter],[xmlRootName])
- › removeValue(name,[private])
- › setHTTPHeader(statusCode,statusText,name,value,charset)
- › setLayout(name)
- › setValue(name,value,[private])
- › setView(name,[noLayout],[cache],[cacheTimeout],[cacheLastAccessTimeout],[cacheSuffix],[layout])
- › valueExists(name,[private])
- › showDebugPanel(boolean)



SAMPLE INTERCEPTOR	SAMPLE PLUGIN
<pre>component{   function configure(){     //configure properties here   }   //method = interception point   function preProcess(interceptData,                       event){   } }</pre>	<pre>component{   function init(){     setPluginName("MyPlugin");     setPluginVersion("1.0");     setPluginDescription("MyPlugin");     setPluginAuthor("YourName");     setPluginAuthorURL("www.my.com");   } }</pre>

COLDBOX CONFIG
<ul style="list-style-type: none"><li>› Coldbox (struct)</li><li>› Settings (struct)</li><li>› Conventions (struct)</li><li>› Environments (struct)</li><li>› loc (struct)</li><li>› WireBox (struct)</li><li>› Debugger (struct)</li><li>› mailSettings (struct)</li><li>› l18n (struct)</li><li>› BugTracers (struct)</li><li>› Webservices (struct)</li><li>› Datasources (struct)</li><li>› LayoutSettings (struct)</li><li>› Layouts (array of structs)</li><li>› Cachebox (struct)</li><li>› Logbox (struct)</li><li>› InterceptorSettings (struct)</li><li>› Interceptors (array of structs)</li><li>› Modules (struct)</li></ul>
CONFIG INJECTED VARS
<ul style="list-style-type: none"><li>› AppMapping</li><li>› Controller</li><li>› LogBoxConfig</li></ul>
SES ROUTE METHODS
<ul style="list-style-type: none"><li>› addRoute(*)</li><li>› addModuleRoutes(pattern, module)</li><li>› setBaseURL()</li><li>› setUniqueURLS()</li><li>› setAutoReload()</li><li>› setExtensionDetection()</li><li>› setValidExtensions()</li><li>› setThrowOnInvalidExtension()</li><li>› setLooseMatching()</li></ul>
* ADDROUTE()
<ul style="list-style-type: none"><li>› [pattern]</li><li>› [handler]</li><li>› [action]</li><li>› [matchVariables]</li><li>› [view]</li><li>› [viewNoLayout]</li><li>› [valuePairTranslation]</li><li>› [constraints]</li><li>› [module]</li></ul>
SES ROUTES
<ul style="list-style-type: none"><li>› Default Pattern :handler/:action?</li><li>› :var ANY Placeholder</li><li>› :var-numeric Numeric</li><li>› :var-alpha Alpha</li><li>› :var{X} Limit by X</li><li>› ? = optional placeholder</li><li>› :regex() Limit by regex</li><li>› Valid Extensions: xml, json, jsont, html, htm, rss</li></ul>
URL ACTIONS (INDEX.CFM?)
<ul style="list-style-type: none"><li>› fwreinit=1 or {reinitPassword}</li><li>› debugmode=true</li><li>› debugpass={DebugPass}</li><li>› fwcache=anything</li></ul>

COLDBOX INTERCEPTION POINTS	CACHEBOX INTERCEPTION POINTS
<ul style="list-style-type: none"><li>› afterConfigurationLoad()</li><li>› afterAspectsLoad()</li><li>› afterPluginCreation() *<ul style="list-style-type: none"><li>› pluginPath, custom, oPlugin</li></ul></li><li>› afterHandlerCreation() *<ul style="list-style-type: none"><li>› handlerPath, oHandler</li></ul></li><li>› afterModelCreation() *<ul style="list-style-type: none"><li>› modelName, oModel</li></ul></li><li>› applicationStart()</li><li>› applicationEnd()</li><li>› onInvalidEvent()<ul style="list-style-type: none"><li>› invalidEvent, override, ehBean</li></ul></li><li>› onException() *<ul style="list-style-type: none"><li>› exception</li></ul></li><li>› onReinit()</li><li>› preProcess()</li><li>› postProcess()</li><li>› preEvent()<ul style="list-style-type: none"><li>› processedEvent</li></ul></li><li>› preLayout()</li><li>› preProxyResults()</li><li>› preRender() *<ul style="list-style-type: none"><li>› renderedContent</li></ul></li><li>› postEvent() *<ul style="list-style-type: none"><li>› processedEvent</li></ul></li><li>› postRender()</li><li>› preViewRender()<ul style="list-style-type: none"><li>› View, cache, cacheTimeout, cacheLastAccessTimeout, cacheSuffix, module</li></ul></li><li>› postViewRender()<ul style="list-style-type: none"><li>› View, cache, cacheTimeout, cacheLastAccessTimeout, cacheSuffix, module, renderedView</li></ul></li><li>› sessionStart()</li><li>› sessionEnd()</li></ul> <p><i>* Has intercept data</i></p>	<ul style="list-style-type: none"><li>› afterCacheElementInsert()<ul style="list-style-type: none"><li>› Cache, cacheObject, cacheObjectKey, cacheObjectTimeout, cacheObjectLastAccessTimeout</li></ul></li><li>› afterCacheElementRemoved()<ul style="list-style-type: none"><li>› Cache, cacheObjectKey</li></ul></li><li>› afterCacheElementExpired()<ul style="list-style-type: none"><li>› Cache, cacheObjectKey</li></ul></li><li>› afterCacheElementUpdated()<ul style="list-style-type: none"><li>› Cache, cacheNewObject, cacheOldObject</li></ul></li><li>› afterCacheClearAll()<ul style="list-style-type: none"><li>› Cache</li></ul></li><li>› afterCacheRegistration()<ul style="list-style-type: none"><li>› Cache</li></ul></li><li>› afterCacheRemoval()<ul style="list-style-type: none"><li>› Cache (name)</li></ul></li><li>› beforeCacheRemoval()<ul style="list-style-type: none"><li>› Cache</li></ul></li><li>› beforeCacheReplacement()<ul style="list-style-type: none"><li>› oldCache, newCache</li></ul></li><li>› afterCacheFactoryConfiguration()<ul style="list-style-type: none"><li>› cacheFactory</li></ul></li><li>› beforeCacheFactoryShutdown()<ul style="list-style-type: none"><li>› cacheFactory</li></ul></li><li>› afterCacheFactoryShutdown()<ul style="list-style-type: none"><li>› cacheFactory</li></ul></li><li>› beforeCacheShutdown()<ul style="list-style-type: none"><li>› cache</li></ul></li><li>› afterCacheShutdown()<ul style="list-style-type: none"><li>› Cache</li></ul></li></ul> <p><i>* Has intercept data</i></p>
SES INTERCEPTOR PROPERTIES	DEPLOY INTERCEPTOR PROPERTIES
<ul style="list-style-type: none"><li>› configFile : relative or absolute routes file defaults to (config/Routes.cfm)</li></ul>	<ul style="list-style-type: none"><li>› tagFile : relative or absolute file location</li><li>› deployCommandObject : instantiation path</li><li>› deployCommandModel : [model name]</li></ul>
ENVIRONMENTCONTROL INTERCEPTOR PROPERTIES	COLDBOXSIDEBAR INTERCEPTOR PROPERTIES
<ul style="list-style-type: none"><li>› configFile : relative or absolute file location</li><li>› fireOnInit : [boolean=true]</li></ul>	<ul style="list-style-type: none"><li>› baseAppPath : [string=cgi.script.name]</li><li>› cssPath : [string]</li><li>› jsPath : [string]</li><li>› URLParamNameList : [string]</li><li>› Yoffset : [numeric]</li><li>› slideSpeed : [numeric]</li><li>› waitTimeBeforeOpen : [numeric]</li><li>› waitTimeBeforeClose : [numeric]</li><li>› isScroll : [boolean]</li><li>› Links : [array]</li><li>› Width : [numeric]</li><li>› visibleWidth : [numeric]</li><li>› imagePath : [path]</li><li>› imageVAlign : [numeric]</li></ul>
SECURITY INTERCEPTOR PROPERTY	RENDERER PLUGIN
<ul style="list-style-type: none"><li>› useRegex : [boolean=true]</li><li>› rulesSource : [string=xml, db, ioc, ocm, model]</li><li>› queryChecks : [boolean=true]</li><li>› preEventSecurity : [boolean=false]</li><li>› validator : [classpath]</li><li>› validatorIOC : [bean name]</li><li>› validatorModel : [model name]</li></ul>	<ul style="list-style-type: none"><li>› renderView(view,cache, cacheTimeout, cacheLastAccessTimeout,cacheSuffix,module,args,collection,co llectionAs)</li><li>› renderLayout(layout,view,module,args)</li></ul>
AUTOWIRE INTERCEPTOR PROPERTIES	
<ul style="list-style-type: none"><li>› completeDIMethodName : [string=onDIComplete]</li><li>› enableSetterInjection : [boolean=false]</li><li>› annotationCheck : [boolean=false]</li><li>› entityInjection : [boolean=false]</li><li>› entityInclude : [string]</li><li>› entityExclude : [string]</li></ul>	



## WireBox

WIREBOX CUSTOM DSL INTERFACE	WIREBOX CUSTOM SCOPE INTERFACE
<pre>Interface{     any init(injector){}     any process(definition,targetObject){} }</pre>	<pre>Interface{     any init(injector){}     any getFromScope(mapping,initArguments){} }</pre>
WIREBOX PROVIDER INTERFACE	WIREBOX PROVIDER METHODS
<pre>Interface{     any get(){}</pre>	<pre>function getXXX() provider="Mapping"{ } }</pre>

WIREBOX BINDER CONFIG
<ul style="list-style-type: none"><li>›logboxConfig (path)</li><li>›cacheBox (struct)</li><li>›scopeRegistration (struct)</li><li>›customDSL (struct)</li><li>›customScopes (struct)</li><li>›scanLocations (array)</li><li>›stopRecursions (array)</li><li>›parentInjector (instance)</li><li>›listeners (array of structs)</li></ul>

WIREBOX SCOPES
<ul style="list-style-type: none"><li>›NOSCOPE OR PROTOTYPE</li><li>›SINGLETON</li><li>›CACHEBOX</li><li>›SESSION</li><li>›APPLICATION</li><li>›REQUEST</li><li>›SERVER</li></ul>

CORE INJECTION DSL
<ul style="list-style-type: none"><li>›Id or model</li><li>›Id:{name}</li><li>›Id:{name}:{method}</li><li>›Provider</li><li>›Provider:{name}</li><li>›Logbox</li><li>›Logbox:root</li><li>›Logbox:logger:{category}</li><li>›Logbox:logger:{this}</li><li>›Wirebox</li><li>›Wirebox:parent</li><li>›Wirebox:eventManager</li><li>›Wirebox:binder</li><li>›Wirebox:populator</li><li>›Wirebox:scope:{scope}</li><li>›Wirebox:properties</li><li>›Wirebox:property:{property}</li></ul>

CACHEBOX INJECTION DSL
<ul style="list-style-type: none"><li>›Cachebox</li><li>›Cachebox:{cache}</li><li>›Cachebox:{cache}:{key}</li></ul>

COLDBOX INJECTION DSL
<ul style="list-style-type: none"><li>›Coldbox</li><li>›Coldbox:setting:{setting}</li><li>›Coldbox:fwSetting:{setting}</li><li>›Coldbox:plugin:{plugin}</li><li>›Coldbox:myplugin:{myPlugin}</li><li>›Coldbox:datasource:{alias}</li><li>›Coldbox:configBean</li><li>›Coldbox:fwConfigBean</li><li>›Coldbox:interceptor:{name}</li><li>›Coldbox:loaderService</li><li>›Coldbox:debuggerService</li><li>›Coldbox:handlerservice</li><li>›Coldbox:interceptorservice</li><li>›Coldbox:moduleservice</li><li>›Coldbox:requestservice</li><li>›Coldbox:pluginservice</li><li>›entityService</li><li>›entityService:{entity}</li><li>›loc</li><li>›loc:{beanName}</li><li>›Javaloader:{class}</li><li>›Webservice:{alias}</li></ul>

WIREBOX OBJECT LIFE CYCLE EVENTS	MAPPING DSL INITIATORS
<ul style="list-style-type: none"><li>›afterInjectorConfiguration()<ul style="list-style-type: none"><li>›injector</li></ul></li><li>›beforeInstanceCreation()<ul style="list-style-type: none"><li>›Mapping, Injector</li></ul></li><li>›afterInstanceCreation()<ul style="list-style-type: none"><li>›Mapping, target, Injector</li></ul></li><li>›afterInstanceInitialized()<ul style="list-style-type: none"><li>›Mapping, target, Injector</li></ul></li><li>›beforeInstanceInspection()<ul style="list-style-type: none"><li>›Mapping, binder, Injector</li></ul></li><li>›afterInstanceInspection()<ul style="list-style-type: none"><li>›Mapping, binder, Injector</li></ul></li><li>›beforeInjectorShutdown()<ul style="list-style-type: none"><li>›injector</li></ul></li><li>›afterInjectorShutdown()<ul style="list-style-type: none"><li>›Injector</li></ul></li></ul>	<ul style="list-style-type: none"><li>›map("name")</li><li>›mapPath("path")</li><li>›mapDirectory("instanation path")</li><li>›with("name")</li></ul>
INJECTOR COMMON METHODS	MAPPING DSL DESTINATIONS
<ul style="list-style-type: none"><li>›Autowire(target,[mapping],[targetID],[annotationCheck])</li><li>›clearSingletons()</li><li>›containsInstance(name)</li><li>›getBinder()</li><li>›getCacheBox()</li><li>›getColdBox()</li><li>›getLogBox()</li><li>›getEventManager()</li><li>›getInstance([name],[dsl],[initArguments])</li><li>›getObjectPopulator()</li><li>›getParent()</li><li>›getScope(scope)</li><li>›getScopes()</li><li>›getVersion()</li><li>›locateInstance()</li><li>›setParent(injector)</li></ul>	<ul style="list-style-type: none"><li>›to("CFC")</li><li>›toJava("java class path")</li><li>›toDSL("dsl string")</li><li>›toFactoryMethod(mapping, method)</li><li>›toRSS("rss URL")</li><li>›toValue(constant value)</li><li>›toWebservice("URI")</li><li>›toProvider("provider mapping name")</li></ul>
OBJECT PERSISTENCE ANNOTATIONS	MAPPING DSL SCOPE PERSISTENCE
<ul style="list-style-type: none"><li>›singleton</li><li>›scope=[scopeName]</li><li>›cache - Cache in CacheBox's default provider</li><li>›cacheBox=[provider]</li><li>›cacheTimeout = minutes</li><li>›cacheLastAccessTimeout = minutes</li></ul>	<ul style="list-style-type: none"><li>›asSingleton()</li><li>›inCacheBox(key,timeout,lastAccessTimeout,provider)</li><li>›Into(scope)</li></ul>
PROPERTY, SETTER, CONSTRUCTOR ARGUMENT ANNOTATIONS	MAPPING DSL CONSTRUCTOR ARGUMENT METHODS
<ul style="list-style-type: none"><li>›Inject=DSL</li><li>›Provider="Mapping"</li></ul>	<ul style="list-style-type: none"><li>›initWith(arguments) - init with specific arguments</li><li>›initArg(name, ref, dsl, value, javaCast)</li></ul>
METHOD ANNOTATIONS	MAPPING DSL SETTER/PROPERTY METHODS
<ul style="list-style-type: none"><li>›Inject=DSL</li><li>›Provider="Mapping"</li><li>›onDIComplete</li></ul>	<ul style="list-style-type: none"><li>›Property(name, ref, dsl, value, javaCast, scope)</li><li>›Setter(name, ref, dsl, value, javaCast)</li></ul>
METHOD CONVENTIONS	MAPPING DSL FACTORY OBJECTS
<p>If an object has the following methods, wirebox will call them for you with the right dependency:</p> <ul style="list-style-type: none"><li>›setBeanFactory( injector )</li><li>›setInjector( injector )</li><li>›setColdBox( coldbox )</li><li>›onDIComplete()</li></ul>	<pre>map("name")     .toFactoryMethod(mapping, method)     .methodArg(name, ref, dsl, value, javaCast) // Repeat as needed</pre>
	REGISITER CUSTOM DSL AND SCOPES
	<p>Register custom DSL and Scopes via mapping DSL</p> <ul style="list-style-type: none"><li>›mapDSL(namespace, path)</li><li>›mapScope(scope,path)</li></ul>
	MAPPING DSL MISCELLANEOUS
	<ul style="list-style-type: none"><li>›asEagerInit() - Not lazy loaded</li><li>›constructor(method) - Override the default constructor of 'init'</li><li>›noAutowire() - Don't autowire object</li><li>›noInit() - Do not call constructor on object</li><li>›onDIComplete(methods) - Register an array of methods for post processing</li><li>›cacheBox(configFile,cacheFactory,enabled,classNamespace)</li><li>›listener(class, properties, name)</li><li>›logBoxConfig( filePath )</li><li>›parentInjector( injector )</li><li>›removeScanLocations( locations )</li><li>›scanLocations( locations )</li><li>›scopeRegistration(enabled,scope,key)</li><li>›stopRecursions( recursionPaths )</li><li>›providerMethod(method, mapping)</li></ul>